CryoEM Skeleton Length Estimation using a Decimated Curve

Andrew McKnight, Kamal Al Nasr, Dong Si, Andrey Chernikov, Nikos Chrisochoides, Jing He

Department of Computer Science
Old Dominion University
Norfolk, VA

{nikos, achernik, jhe}@cs.odu.edu

***Abstract* - Cryo-electron Microscopy (cryoEM) is an important biophysical technique that produces 3-dimensional (3D) images at different resolutions. De novo modeling is becoming a promising approach to derive the atomic structure of proteins from the cryoEM 3D images at medium resolutions. Distance measurement along a thin skeleton in the 3D image is an important step in de novo modeling. In spite of the need of such measurement, little has been investigated about the accuracy of the measurement in searching for an effective method. We propose a new computational geometric approach to estimate the distance along the skeleton. Our preliminary test results show that the method was able to estimate fairly well in eleven cases. This method is also able to detect outliers in the Electron Microscopy Data Bank in which the loops do not match the skeletons very well.**

Skeleton, computational geometry, electron microscopy, protein, length

I. Introduction

Electron cryo-microscopy (cryoEM) continues to produce 3-dimensional (3D) images of large protein complexes with a wide range of resolutions from about 3Å to over 80Å. However, only an extremely small portion of such 3D images has been resolved to the atomic structures [1]. *De novo* modeling has been demonstrated as a promising method to derive atomic protein structures from a 3D image at the medium resolution, such as 5-10Å [2-4]. D*e novo* modeling is a general method that does not rely on the availability of a template structure in the Protein Data Bank (PDB). Figure 1 illustrates the major steps in *de novo* modeling to generate an initial atomic structure. The input information includes the 3D image, called the protein density map, and the amino acid sequence of the protein. It uses image processing methods to detect the secondary structures such as helices (red sticks in Figure 1) and β-sheets in the 3D image [5-10]. It uses skeletonization methods to detect the skeleton connection (green in Figure 1) in the 3D image [4, 11]. It uses secondary structure prediction tools to determine their location on the protein sequence [12-15]. The topology of the secondary structures can be inferred by combining the two sources of information about the secondary structures, one from the 3D image and the other from the protein sequence [2, 16]. Once the topology of the secondary structures is determined, the atomic structure will be modeled.

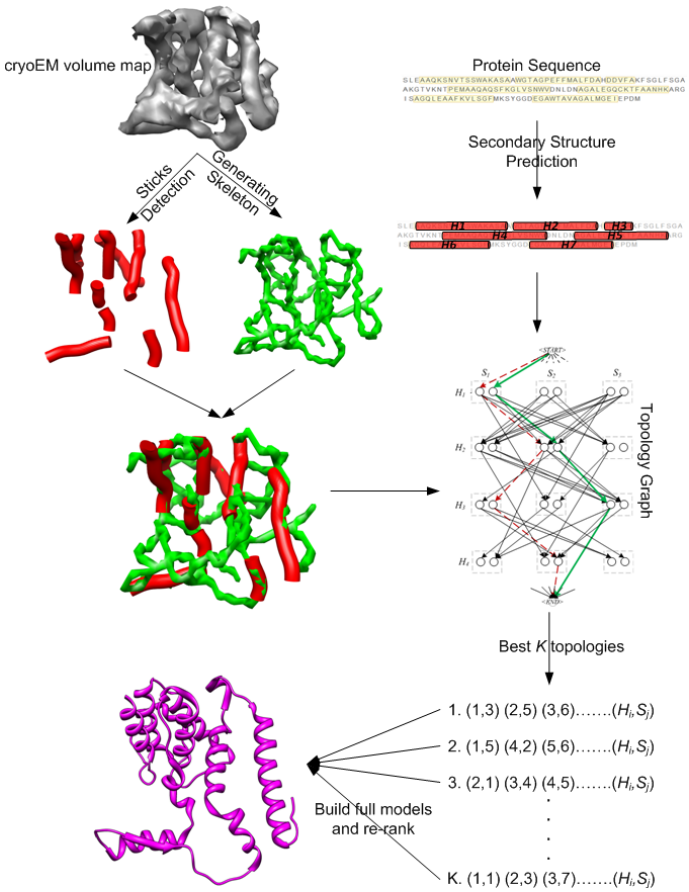

Figure 1: The *de novo* modelling approach. The two inputs are the protein SSEs on the sequence and the 3D image (density map) at medium resolution. The helices (red sticks) were detected using *SSELearner* [10] and the skeleton (green) was detected using Gorgon [4]. The protein structure is shown as a ribbon (purple).

A protein sequence has a direction, from the N-terminal to the C-terminal. When the protein sequence is threaded in the 3D image, it visits the detected helix sticks in a unique order. The determination of this order is the topology problem. The topology of the Secondary Structure Elements (SSEs) is a critical piece of information in protein structure prediction. It

has been demonstrated that the topology of the helix sticks can be derived using graph-matching approaches[16] [2]. Our previous work translated the graph-matching problem into a constrained shortest path problem in a topology graph. We presented a dynamic programming algorithm in $O(N^2 2^N)$ time [2]. The idea of topology determination is to use the distances between the SSEs as a metric in matching. For example, two helices 5Å apart in the 3D image should be matched to two helices of similar distance in the protein sequence. The distance between two helices on the amino acid sequence can be translated into the distance in 3D space by assuming a 3.8Å distance between two consecutive amino acids on the sequence. The assumption is that the correct topology results in the overall best match when all pairs of helices are considered.

Skeletonization of the 3D image has been shown to be an important technique to extract the connections between the SSEs. The skeleton can be derived using thinning and pruning methods [17]. It roughly represents the major paths in the 3D image. The skeleton is a set of grid points, or voxels, along the paths that appear to zigzag. Ideally, the distance between two specific ends of two helices should be measured along the skeleton connecting the two ends. If we simply add the length of the line segments along the path, there is a danger of over estimation due to the potential zigzag nature of the path. Moreover, the skeleton is expected to contain errors, since the 3D image often contain errors. It is not clear if the skeleton length estimation methods are accurate enough for topology determination.

This paper introduces a new method to approximate the distances between the detected SSEs from a density map, for use in the matching algorithm. In particular, we measure the length along the skeleton using a combination of graph-theoretic and computational geometric methods. We tested the method using a small dataset consisting of the experimentally derived 3D images from the Electron Microscopy Data Bank. The measured length appears to agree with the expected length when the atomic structure of the turn aligns well with the skeleton. In our future work we expect to perform an extensive evaluation of the algorithm and fine tune it accordingly.

II. Methodology

Problem: *estimate the skeleton length connecting two helices in the 3D image*.
The solution, which is an approximation, is described in the flow chart in Figure 2. There are three basic steps: (1) pre-process the skeleton, (2) construction of the trees and paths, and (3) decimation of the paths derived from step (2).

A. Processing of the Skeleton

First, we apply the skeletonization method using Gorgon [4]. An example of the skeleton is shown in green in Figure 1. The skeleton produced by Gorgon and includes the helix region. In order to estimate the length of the portion of the skeleton that corresponds to the turn of the atomic structure, we need to mask out the portion within the helix.

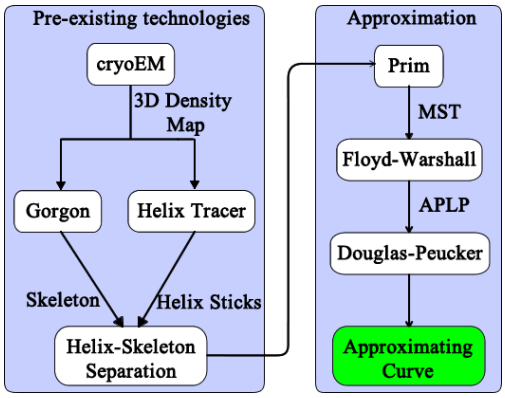

Figure 2: Algorithmic flow of the different components to calculate the piecewise linear approximation.

In theory, we can use a tool, such as *SSELearner*, to detect the position of the helices [10]. In this work, we used the true location of the helices obtained from the PDB file. Since this is the initial test about the estimation of the skeleton length, we hope to test the accuracy when the helices are accurately detected. We plan to use the detected helix positions in the future. We removed the skeleton that falls in a cylinder of 5Å in diameter at the helix. The centers of the two ending circles are determined by the geometrical center of the first three and the last three C_α atoms on the helix. After removal, what is left is the set of skeleton voxels corresponding to the protein backbone not found in any secondary structures. These parts of the backbone are the turn / loops that connect two adjacent helices on the sequence

B. Graph Theoretic Approach

Virtually every approximating method requires that the input data points be ordered in some way. However, the skeleton points are those grid points (voxels) in 3D without any order. Our first step is to construct a connected graph of the skeleton voxels. Then we construct the minimal spanning tree (MST) using Cormen's implementation of Prim's algorithm [18]. Without loss of generality we use arbitrary edges of the MST to describe the path of the turn, so we must throw out the outlying branches and create a piecewise linear curve. However, we need to eliminate the minimum amount of data points to preserve as much information as possible. In addition, we determine this path without any other inputs, such as helix endpoints, because they can lie some distance inside the helix area, spuriously adding length to the approximation. To find the path in this way, we use the Floyd-Warshall algorithm (again implemented by Cormen in [18]) to compute all-pairs shortest paths in the MST, and reconstruct the longest such path, which we refer to as the all-pairs longest [simple] path (APLP). Conveniently, the APLP implies an order on the points it contains for use in the actual approximating step.

C. *Computational Geometric Approach*

An artifact directly related to the initial skeleton construction is that our APLP contains right angles at the skeleton voxels, giving it the undesired zigzag appearance. This introduces a margin of error in length when compared to the relatively smooth curve of the protein backbone. To overcome this, we

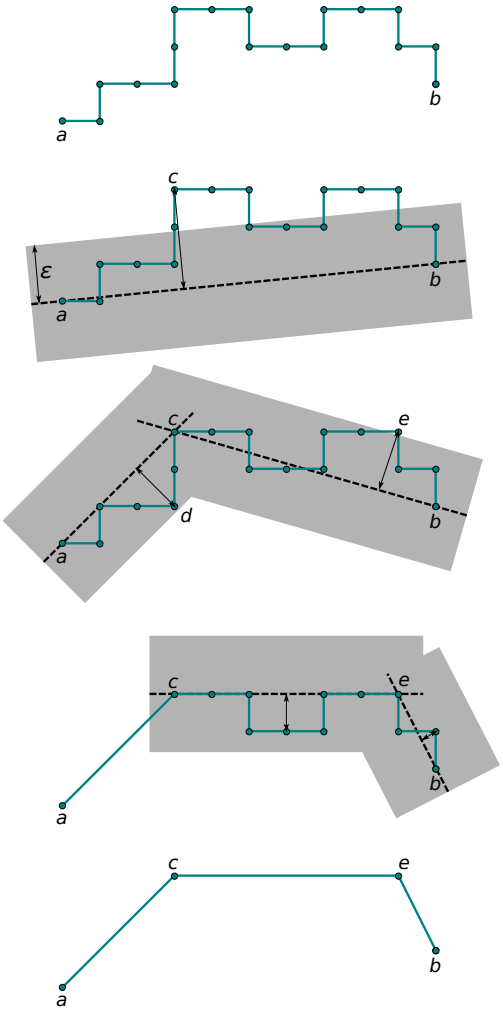

Figure 3: Illustration of the Douglas-Peucker polyline decimation algorithm at work in a 2D case.

simplify the line by removing certain points using the Douglas-Peucker line simplification, generalized to three dimensions by modifying de Halleux's implementation given in [19].

The Douglas–Peucker line decimation algorithm [6] allows us to remove points from a piecewise-linear three-dimensional line (referred here as polyline), such that the resulting polyline remains within some tolerance epsilon ε from the original one. Consider a two-dimensional example in Figure 3. The top drawing shows an initial polyline a...b. Its points are chosen from a rectilinear grid, and therefore the total length of the polyline a...b overestimates a smoother line that could connect points a and b and pass through the same geometrical neighborhood. The algorithm is recursive, and takes as parameters the tolerance ε and a multi-point segment of a polyline (which is initially the original polyline). At each recursive call it finds an interior point of the current segment which is the most distant from the straight line connecting the end points of the segment. If the most distant point is within ε from the straight line, the segment is replaced by the straight line, and all interior points are removed. Otherwise, the segment is split into two sub-segments by this most distant interior point, and the algorithm proceeds recursively on each of the sub-segments. The example in the Figure shows how the initial polyline a...b is simplified into polyline aceb. Figure 4 shows the result of decimating an APLP from a test case in three dimensions with ε = 1.0.

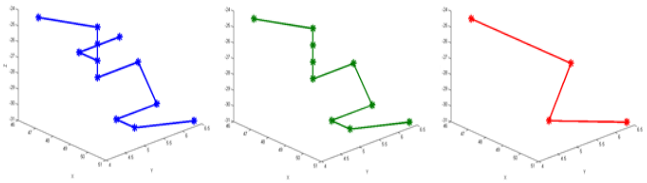

Figure 4: The MST (blue), APLP (green) and decimate curve approximation (red) for a 3-residue turn in EMDB_5001.

III. RESULTS

We used the 3D images from EMDB and their corresponding atomic structures from PDB to test our method. The 3D images in EMDB are experimentally derived, and provide test cases of the real experimental data. We selected five density maps from EMDB with different resolutions. These data include EMDB5030_6.4Å, EMDB1733_6.8Å, EMDB_5001_4.2 Å, EMDB1740_6.8 Å, and EMDB_5168_6.6 Å. Each of these 3D images is aligned with their PDB structures at download. We extracted turns less than seven amino acids in length from the PDB file and extracted the corresponding local regions around the turn connected by two helices. We obtained the skeletons using Gorgon and processed the skeleton so that those inside the cylinder of the helices are deleted leaving the portion belonging the turns. We measured the length of the processed skeleton and compared it with the expected length of the turn. The expected length of the turn is calculated by the number of the amino acids on the turn with the consideration of 3.8Å in between two amino acids.

A. **ε** *Threshold*

ε is one of the major parameters in the Douglas-Peucker algorithm affecting the approximated skeleton length between two helices. In general**,** the smaller the ε value, the less change in the decimated curve compare to the original one. Figure 5 shows the skeleton length measured using different ε values in the range [0.5, 3.5]. The length corresponds to the 3-residue turn in EMDB 1733. In this case, ε = 0.75 produces the closest approximation to the actual loop length (see case 3 in Table 1). For all the cases in our current test, a value of ε between

0.5 and 1.0 produces an approximated length very near to the expected length of the turn, as illustrated in Figures 6 and 7.

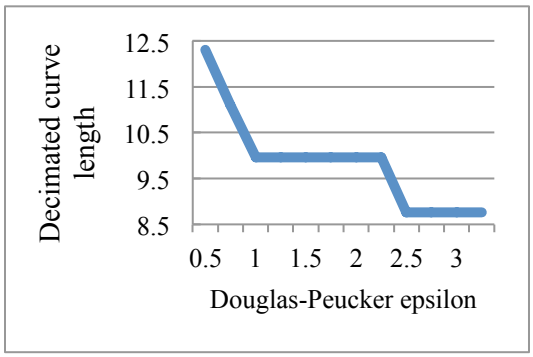

Figure 5: ε values of the Douglas-Peucker method.

Figure 6 shows an example of decimated curve for the 3-residue turn (row 4 of Table 1). The skeleton derived from Gorgon was shown as the surface representation using Chimera [20]. The skeleton was superimposed on the backbone C_α trace of the protein, obtained from the PDB file. In this case, the backbone chain appears to fit in the skeleton fairly well at the turn region (green). The central end point (yellow) of the helix was estimated using the geometrical center of the last three C_α atoms on the helix. After the removal of the helix portion of the skeleton, the MST was built to find the APLP (purple). The skeleton length corresponding to the turn was estimated using the decimated curve (dark blue) to be 11.11Å. It is fairly close to the estimated distance of 11.4Å.

Table 1 summarizes the testing results. The length-3 turns dominate with eight out of eleven test cases. Somehow the length-3 turns are more popular than other kinds of turns among the helix-turn-helix motif. If we use the ε value (column 4 of Table 1) that produces the closest estimation with respect to the true length, the approximated skeleton length (column5) is fairly close to the expected length with the difference between 0.14Å to 1.32Å (column 7). This result suggests that it is possible to estimate the length of the turn using the skeleton length at least for the length-3 turns. These results support the previous finding of our group and other groups in terms of the use of graph matching for topology determination. Our preliminary test here shows that there is a ε value that produces close estimation of the skeleton length for certain type of turns. When the turn is even shorter, having one amino acid, our current estimation still gives reasonable accuracy (row 1 and 2 of Table 1). However, the ε value varies more in the case of length-3 turns. We need more test cases for the other lengths to make a conclusion.

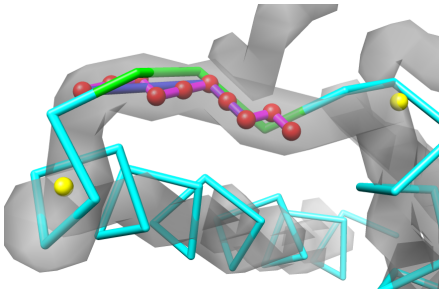

Figure 6: The decimated curve (dark blue) for ε = 0.75 for the 3-residue loop in EMDB 1733, along with the APLP (purple), skeleton surface (grey) and its voxels (red), turn (green), helices (cyan) and estimated helix endpoints (yellow).

Table 1: Results of approximation.

No	ID[a]	AA[b]	ε[c]	Approx[d]	Real[e]	Diff[f]
1	5138	1	0.5	5.32	3.8	1.52
2	5030	1	1.0	4.077	3.8	0.277
3	1733	3	0.5	10.95	11.4	0.45
4	1733	3	0.75	11.11	11.4	0.33
5	5001	3	0.75	11.88	11.4	0.48
6	5001	3	0.5	11.66	11.4	0.26
7	5030	3	0.5	11.34	11.4	0.14
8	5030	3	0.75	11.73	11.4	0.33
9	5001	3	0.5	12.72	11.4	1.32
10	5168	4	0.5	15.2	6.92	8.28
11	1740	6	0.5	21.89	22.8	0.91

a: EMDB ID.
b: Number of amino acids in the turn.
c: Douglas-Peucker epsilon value used to derive the approximating curve.
d: Actual length of the turn. For a turn with *n* residues, the length *l* is assumed to be $l = n * 3.8$Å.
e: Approximated skeleton length
f: Difference between actual and approximated lengths.

The cases in Table 1 (except case 10) all have a skeleton that aligns well with the actual protein backbone throughout the loop. However, in some cases we see that the skeleton is not aligned well with the backbone chain of the loop. Sometimes it lies inside the loop, producing a shorter approximated length than the expected length. The skeleton for case 10 lies inside the actual protein backbone (Figure 7), leading to the large approximation error. Further study on the relationship between distances of skeletons to backbones and the approximated lengths is needed, as well as investigation of skeletonization, which is outside the scope of this paper.

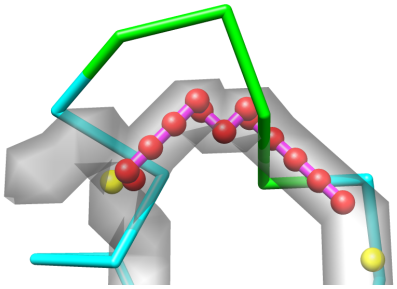

Figure 7: An: example of mis-aligned skeleton and the backbone of the turn (case 10 from Table 1), producing an erroneously short approximation of the loop length.

IV. Summary

We have investigated the question how accurate it can be to estimate the skeleton length between two helices. Although the skeleton length has been used in topology determination, there has not been a detailed study in the computation of a 3D curve that closely approximates the skeleton of the image. We propose an effective method in estimating the skeleton length using a decimated curve. A test of eleven cases using the experimentally derived data shows that the estimation can be potentially accurate to a fair degree if the backbone of the protein chain fits in the skeleton. This was demonstrated well for the helix-turn-helix motif with three amino acids on the turn. Our method can detect the turns in which the turn is outside the skeleton. We plan to carry further investigation in this direction.

Acknowledgment

This work is funded in part by NSF grants: CCF-1139864, CCF-1136538, and CSI-1136536 and by the Richard T. Cheng Endowment, the ODU MSF fund and the ODU startup fund.

References

1. Lawson, C.L., et al., *EMDataBank.org: unified data resource for CryoEM.* Nucleic Acids Res, 2011. **39**(Database issue): p. D456-64.
2. Al Nasr, K., et al., *Ranking valid topologies of the secondary structure elements using a constraint graph.* J Bioinform Comput Biol. **9**(3): p. 415-30.
3. Lindert, S., et al., *EM-fold: De novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps.* Structure, 2009. **17**(7): p. 990-1003.
4. Baker, M.L., et al., *Modeling protein structure at near atomic resolutions with Gorgon.* Journal of Structural Biology, 2011. **174**(2): p. 360-373.
5. Jiang, W., et al., *Bridging the information gap: computational tools for intermediate resolution structure interpretation.* J Mol Biol, 2001. **308**(5): p. 1033-44.
6. Del Palu, A., et al., *Identification of Alpha-Helices from Low Resolution Protein Density Maps.* Proceeding of Computational Systems Bioinformatics Conference(CSB), 2006: p. 89-98.
7. Baker, M.L., T. Ju, and W. Chiu, *Identification of secondary structure elements in intermediate-resolution density maps.* Structure, 2007. **15**(1): p. 7-19.
8. Kong, Y., et al., *A Structural-informatics approach for tracing beta-sheets: building pseudo-C(alpha) traces for beta-strands in intermediate-resolution density maps.* J Mol Biol, 2004. **339**(1): p. 117-30.
9. Zeyun, Y. and C. Bajaj, *Computational Approaches for Automatic Structural Analysis of Large Biomolecular Complexes.* Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 2008. **5**(4): p. 568-582.
10. Si, D., et al., *A machine learning approach for the identification of protein secondary structure elements from electron cryo-microscopy density maps.* Biopolymers. **97**(9): p. 698-708.
11. Ju, T., M.L. Baker, and W. Chiu, *Computing a family of skeletons of volumetric models for shape description.* Comput Aided Des, 2007. **39**(5): p. 352-360.
12. McGuffin, L.J., K. Bryson, and D.T. Jones, *The PSIPRED protein structure prediction server.* Bioinformatics, 2000. **16**(4): p. 404-5.
13. Ward, J.J., et al., *Secondary structure prediction with support vector machines.* Bioinformatics, 2003. **19**(13): p. 1650-5.
14. Pollastri, G. and A. McLysaght, *Porter: a new, accurate server for protein secondary structure prediction.* Bioinformatics, 2005. **21**(8): p. 1719-20.
15. Pollastri, G., et al., *Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles.* Proteins, 2002. **47**(2): p. 228-35.
16. Abeysinghe, S. and T. Ju. *Shape modeling and matching in identifying protein structure from low resolution images*. in *Proceedings of the 2007 ACM symposium on Solid and physical modeling* 2007. Beijing, China.
17. Ju, T., Matthew L. Baker and Wah Chiu, *Computing a family of skeletons of volumetric models for shape description.* Computer Aided Design, 2007. **39**(5): p. 8.
18. Cormen, T.H. *Java files for the com.mhhe.clrs2e package,"* 2003 [cited 2012 July 12]; Available from: http://classes.engr.oregonstate.edu/eecs/winter2005/cs325/clrs2e/.
19. De Halleux, J. *A C++ implementation of Douglas-Peucker line approximation algorithm*. 2009 [cited 2012 August 4]; Available from:

http://www.codeproject.com/Articles/1711/A-C-implementation-of-Douglas-Peucker-Line-Approxi.

20. Pettersen, E.F., et al., *UCSF Chimera—A visualization system for exploratory research and analysis.* Journal of Computational Chemistry, 2004. **25**(13): p. 1605-1612.